10/648067

Hit List

First Hit Clear Generate Collection Print Fwd Refs Bkwd Refs Generate OACS

Search Results - Record(s) 1 through 9 of 9 returned.

☐ 1. Document ID: JP 2004258406 A

L12: Entry 1 of 9 File: JPAB Sep 16, 2004

PUB-NO: JP02004258406A
DOCUMENT-IDENTIFIER: JP 2004258406 A
TITLE: SYSTEM AND METHOD FOR TEXT SPEECH SYNTHESIS

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

☐ 2. Document ID: NB940497

L12: Entry 2 of 9 File: TDBD Apr 1, 1994

TDB-ACC-NO: NB940497

DISCLOSURE TITLE: Variable Expiration Dates on a Specific Executable

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

☐ 3. Document ID: NN9211102

L12: Entry 3 of 9 File: TDBD Nov 1, 1992

TDB-ACC-NO: NN9211102

DISCLOSURE TITLE: Method of Applying, Monitoring and Recording Multiple Test Criteria.

4. Document ID: NN9207308

L12: Entry 4 of 9 File: TDBD Jul 1, 1992

TDB-ACC-NO: NN9207308

DISCLOSURE TITLE: APL2 Calling C Routines that use Reference Parameters.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

5. Document ID: NN85046584

L12: Entry 5 of 9 File: TDBD Apr 1, 1985

TDB-ACC-NO: NN85046584

DISCLOSURE TITLE: Insuring Insert Integrity Over Variable-Length, Spanned Records

SECURITY: Use, copying and distribution of this data is subject to the restictions in the Agreement For IBM TDB Database and Related Computer Databases. Unpublished - all rights reserved under the Copyright Laws of the United States. Contains confidential commercial information of IBM exempt from FOIA disclosure per 5 U.S.C. 552(b)(4) and protected under the Trade Secrets Act, 18 U.S.C. 1905.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

6. Document ID: NN84014363

L12: Entry 6 of 9 File: TDBD Jan 1, 1984

TDB-ACC-NO: NN84014363

DISCLOSURE TITLE: Eavesdrop-Detecting Quantum Communications Channel

SECURITY: Use, copying and distribution of this data is subject to the restictions in the Agreement For IBM TDB Database and Related Computer Databases. Unpublished - all rights reserved under the Copyright Laws of the United States. Contains confidential commercial information of IBM exempt from FOIA disclosure per 5 U.S.C. 552(b)(4) and protected under the Trade Secrets Act, 18 U.S.C. 1905.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☐ 7. Document ID: WO 2006036958 A1, US 20060074971 A1

L12: Entry 7 of 9 File:

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☐ 8. Document ID: CN 1380627 A

L12: Entry 8 of 9 File: DWPI Nov 20, 2002

DERWENT-ACC-NO: 2003-185204
DERWENT-WEEK: 200319
COPYRIGHT 2007 DERWENT INFORMATION LTD

TITLE: Credit card illegal use resisting system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☐ 9. Document ID: DE 3720427 A, FR 2600444 A, US 4839792 A

L12: Entry 9 of 9 File: DWPI Dec 23, 1987

DERWENT-ACC-NO: 1988-000669
DERWENT-WEEK: 198801
COPYRIGHT 2007 DERWENT INFORMATION LTD

TITLE: Terminal integrated circuit card for bank or store - has CPU and memory which stores data trains from integrated circuit handling system, readable using CPU

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L10 or L11	9

Display Format: [-] Change Format

L12: Entry 1 of 9 File: JPAB Sep 16, 2004

PUB-NO: JP02004258406A
DOCUMENT-IDENTIFIER: JP 2004258406 A
TITLE: SYSTEM AND METHOD FOR TEXT SPEECH SYNTHESIS

PUBN-DATE: September 16, 2004

INVENTOR-INFORMATION:

NAME	COUNTRY
MURASE, KENTARO	
KATAE, NOBUYUKI	
WATANABE, KAZUHIRO	

ASSIGNEE-INFORMATION:

NAME	COUNTRY
FUJITSU LTD	

APPL-NO: JP2003049917
APPL-DATE: February 26, 2003

INT-CL (IPC): G10L 13/06; G10L 13/00; G10L 13/08

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a system and a method for text speech synthesis that generate a meter of high quality by reducing deterioration in naturalness of a meter pattern due to an error in modification analysis.

SOLUTION: The text speech synthesis system which inputs a text character string, analyzes language information regarding the text character string, generates a metric pattern according to the language information, and synthesizes a speech waveform according to the language information and meter pattern, finds a morpheme string and grammar information by taking a morpheme analysis of the text character string, constitutes a phrase according to the morpheme string, finds information regarding modifications between phrases, sets reliability of information regarding modification between the phrases according to the morpheme string or/and the information regarding modification between the phrases, and updates the meter pattern according to the reliability of the information regarding the modification.

L12: Entry 2 of 9 File: TDBD Apr 1, 1994

TDB-ACC-NO: NB940497

DISCLOSURE TITLE: Variable Expiration Dates on a Specific Executable

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, April 1994, US

VOLUME NUMBER: 37
ISSUE NUMBER: 4B
PAGE NUMBER: 97 - 98

PUBLICATION-DATE: April 1, 1994 (19940401)

CROSS REFERENCE: 0018-8689-37-4B-97

DISCLOSURE TEXT:

A program is disclosed that allows the shipping of an executable program to a customer with an modifiable expiration date. This invention allows distribution of an ASCII file that can be updated by the user (with the correct information provided by the programmer) without having to update the executable program. - Currently companies derive revenue from shipping their executables with a license option such that if the license is expired (past the date allowed to execute) a message is printed, and the program does not execute. An implementation of this concept is known as a timebomb, and this may involve a hardcoded date compiled into the executable, making use of a license server that checks a file to determine if the user and/or the license is valid, or having the executable check the file directly. There is a need for a light weight license encoding strategy for tools that are to be distributed to customers that avoids compiling in the date (which would force the reshipping of the executable when the expiration date has passed), and avoids forcing the customer to purchase extra license server software. The lightweight solution to this problem is as follows: Ship an ASCII file that contains three fields per line. The fields are the executable name, the date that the program will expire on, and an encrypted representation of that date. The encrypted date is the the number of seconds since the creation of UNIX* (known as the epoch), converted into the equivalent ASCII digits, and a different prime number is added to each digit so that a repeatable pattern is not discernable. This ensures that the encrypted representation of the date is printable and that the new file can be faxed or sent via e-mail to the customer without necessitating the resending of the executable. - Decoding of the encrypted expiration date is done with each invocation of the executable. The encrypted date is first read in from the file, the series of prime numbers is then subtracted from the associated digit in the encrypted date, the number is converted from ASCII to an integer and is finally compared against the epoch time currently registered on the machine running the application. If the epoch is less than or equal to the encrypted date, then the application continues execution, if the encrypted date is less than the epoch, then a message is generated stating that the license has expired. - The user can then contact the supplier for a new license file which will contain a new encrypted expiration date and continue to use the application until such a time that the new encrypted date is reached. Using this concept, the seller can control who is using the application and also receive royalties each time the license must be renewed. -

The unique part of this invention is the lightweight encoding scheme for the encrypting/decrypting expiration dates. An example of how it would work is as follows: The epoch time for expiration is: 74002345 Generate an ASCII string to represent that integer. Create an array of prime numbers. Index the ASCII representation of the epoch by individual bytes and add the prime number that is at that same index. - Using the string above, the first byte is a '7' (which has an ASCII numeric value of 37) and add the first prime number in the sequence that you generated (an example sequence would be 2 3 5 7 11 13 17 19). The resulting output for the first byte in the encoded string would be '9' (which has an ASCII value of 39 - the 37 of the '7' plus the first prime number in the sequence, 2). Encrypting the epoch time from above using this algorithm would result in: 9757=@EH Decoding this encrypted string would be a matter of subtracting off each of the prime numbers in the sequence above from the appropriate byte and then converting the ASCII representation into an integer. Compare this integer to the current epoch time and determine whether or not the license has expired. - Since the ASCII file is editable by the customer, the data integrity of the encrypted date is not guaranteed. This means that a customer may try to enter their own encrypted date - and they may be able to generate a string that passes the test. To add some more security to this encryption, check the decrypted epoch time against the current epoch plus a constant number of seconds (one year, for example). If the decrypted epoch exceeds that arbitrary constant, then the application can be pretty sure that the encrypted epoch has been altered by the user. - The integrity of the license can also be affected if the user of the software modifies the system clock and resets it to some date prior to the expiration of the license. There is no way to protect against this, so this can be seen as a way to work around the expiration of the license * Trademark of UNIX System Laboratories.

Previous Doc Next Doc Go to Doc#

L12: Entry 3 of 9 File: TDBD Nov 1, 1992

TDB-ACC-NO: NN9211102

DISCLOSURE TITLE: Method of Applying, Monitoring and Recording Multiple Test Criteria.

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, November 1992, US

VOLUME NUMBER: 35
ISSUE NUMBER: 6
PAGE NUMBER: 102 - 114

PUBLICATION-DATE: November 1, 1992 (19921101)

CROSS REFERENCE: 0018-8689-35-6-102

DISCLOSURE TEXT:

- Described is a method of applying, monitoring and recording multiple test criteria for the efficient simultaneous characterization of performance envelope data for multiple units under test. - The multiple test criteria method provides an optimized approach for estimating multiple boundary conditions along multiple independent unidirectional vectors where each test is applied empirically and simultaneously to all units under test. Through the use of this optimized testing method, stress data can be obtained quickly, thereby enabling the user to evaluate design capabilities. - In prior art, manual and/or exhaustive testing has been used to determine the operating space or the performance envelope for the unit under test. This envelope was determined by empirically applying test criteria to the unit under test while various stress parameters were applied. Mapping the pass or fail response of the unit under test was then performed in order to obtain an estimate of the pass/fail boundary of the product. Typically, this procedure was performed manually on one unit at a time so as to apply one or more test criteria sequentially. As automation became evident, the manual one at a time approach became obsolete. Software was developed which exhaustively maps the response of each unit under test at each stress condition. However, this approach was considered inefficient since the brute-force algorithm led to many meaningless applications of the test and was time consuming. Also, this approach lacked the means to obtain separate data for different test criteria. For example, if separate performance envelopes were being obtained for a product running test criteria A and results were also required for test criteria B, the entire test had to be run separately for each of these test criteria. - The implementation of the multiple test criteria method consists of the disclosed algorithms used to control automated test equipment. The algorithms make use of the stress vector concept where the vector involves a combination of stresses. A vector may be thought of as a stress parameter where the controlled setting is under the control of an algorithm. The vector is incremented in predetermined steps by the algorithm, and all chosen criteria are applied until all units under test fail their criteria. - The following list outlines the features provided in the multiple test criteria method: - Selectable Number of Pass/Fail Retries - The user is able to select any number of pass or fail events which must occur in a row in order for the result to be considered valid. This allows the filtering of intermittent or spurious results. -

- Selectable Combination of Search Types - The user may select one of several search algorithms which include linear searches, binary searches and a combination of linear/binary and binary/linear searches. The combination searches use one search algorithm for the gross portion of the test and another for narrowing down the pass/fail boundary, thereby enabling the boundary to be located with greater precision. The relative effectiveness of the various search types depends on many factors, including the number of samples tested, the number of test criteria and the expected consistency of the results. Providing flexibility in the search types allows the user to select the most efficient search for the application. - - Return to Nominal for Detection of Multiple Failure Points - The algorithm allows for a return to normal situation in which the stress factors are optionally returned to nominal values until the case of interest is ready to run. This allows the determination of many boundary points beyond the initial failing point where a simple algorithm would not allow the test to continue. For example, it may be desirable to complete a power on self test (POST) prior to the execution of diagnostic test cases under the control of an operating system. - - Simultaneous Characterization - Because the results for all test criteria are determined simultaneously, the results are not affected by external factors, such as time of day, variances in stress equipment, etc. Therefore, results can be compared with confidence that any differences noted are significant. - - Expandability - The algorithms can be applied to any number of samples with any number of test criteria in use. This improves the efficiency as the complexity of the test scenario increases. - - Advanced Test Data Storage, Updating and Output Features - Because actual measured data is obtained in real time rather than a pass or fail indication for predetermined stress points, the output flexibility is increased. This includes the following: - Detection and Differentiation of Multiple Failure Modes - Because both reported and hangup failures are detected and tracked separately, a better understanding of the failure is attained. - - Separate Data for Last Pass and First Fail Data - As the test proceeds, the worst-case pass data and the best-case fail data for each test criteria will be constantly updated. This allows retrieval of data from interrupt or incomplete tests and for graphical display of tests in progress. - - Multiple Output Formats - The final output data can be made available in several forms. Among these are the summary file form which contains the data in a form easily interpreted by a human operator and the envelope file form which contains the data in a form which may be input directly to a spreadsheet or graphing program. - The multiple test criteria method is considered applicable in other areas, such as: a) iterative mathematical approaches in which the operation is performed repeatedly until a certain condition is met; b) in data searches in which a segment of data is analyzed for the presence of one or more specific strings or values; and c) in automated mechanical tests such as hardness, breaking stress or tolerance testing. Additionally, the data storage format approach could be applicable in any case where many variables must be updated to reflect the state of many objects. The algorithm is equally useful for testing products at any level of integration. The flowcharts indicate a unit under test reset by the term POR for power on reset. For different products, another type of reset might be used. The term hangup refers to the detection of the condition in which the unit under test has not indicated a failure or halted during the running of a test case, but in which the unit under test has not reported or responded within the expected amount of time. This is useful in the case where a product hangs or locks up during the test when the hang cannot be attributed to the execution of a diagnostic test. - The flow charts (Figs. 1 - 10) show the various steps required by the multiple test criteria method during the processing of the algorithms. Fig. 1 shows a flow chart of the steps required to run the high-level run-time algorithm. The remaining flow charts, Figs. 2 to 10, show the steps to run the algorithms of each main sub-process. The best way to follow the flow charts is to visualize several of the units under test. For example, to determine the limits to which the system can be stressed before failures occur requires the determination of the exact failure point at which each diagnostic test begins to fail on each of the units under test as the stress increases. The descriptions of the various flow charts are as follows: Fig. 1 - Runtime.Flo - This is a global process. At the completion of this

process, all equipment is initialized. A complex series of events now occurs. In general, the method sets a stress condition, applies a test criterion, logs the results of the test and then determines the next stress condition to be set based on the results of the test. The algorithm allows simultaneous determination of many criteria for many systems. This process continues until all desired data is obtained, after which all equipment is shut down. Within the global process are several important sub-processes. Each of these is shown in its own flow chart (Figs. 2 to 10) and described below. The descriptions are simplified in that many additional checks and adjustments are made during the course of each process. Fig. 2 - INITVAR.FLO - In this sub-process, all data arrays are set up and initialized and the starting point is determined. Fig. 3 - INITVEC.FLO - In this sub-process, the initial stress point is programmed into the test equipment and verified. Then the units under test are started. Fig. 4 - PROCLOOP.FLO - In this looping sub-process, each unit under test is examined in sequence. One of several possible statuses is determined for each unit under test. If a system is determined to have failed a test, this information is logged. The loop continues until the status of all tests on all units under test have been determined. Fig. 5 - PROCPAR.FLO - In this sub-process, communications from the units under test are interpreted and reacted to. The units under test can be sending one of three types of messages to the control system. The first two of these are the messages that a particular test passed or failed. When either of these occurs, the data points associated with that test on that unit under test are updated. The other possible message is a request for permission to run a particular test. When this occurs, the control system consults several rules, taking into account whether the operator has selected that test, whether that test has failed previously, etc., and responds appropriately to the unit under test. This sub-process also tracts and reacts to the status unit under test, such as if each system has completed its test, then the next step is initiated. Fig. 6 - CHKDONE.FLO - This process determines whether a particular test vector has been completed for any units under test and returns its decision to the main processor. At each invocation, it determines if any unit under test can be eliminated from future tests. When all units under test have been eliminated, the vector is complete. Fig. 7 - NEXTSTEP.FLO - This sub-process resets all data arrays and prepares the test system to run the tests at the next stress point. Figs. 8 & 9 - RECOVER.FLO/RECNOM.FLO - These sub-processes occur when a unit under test has hung. They conditionally bring the stress conditions back to nominal and reset the unit under test, allowing it to recover from the hang. Fig. 10 - WAITPOST.FLO - This is a special sub-process which occurs between the time that the units under test are initialized and the time that they report that they have successfully brought themselves up. Hangups or failures during initialization are detected here. - Not shown in the flow charts are the following periodic checks, all of which halt the algorithms upon failure: - Every five seconds, read the status of all connected equipment, such as temperature chambers and power supplies. Halt upon report of abnormal status. - Every five seconds, read the current values from all connected equipment and compare each value read with the acceptable range of equipment. Halt if any parameter exceeds the acceptable equipment ranges. - Every sixty seconds, read all data logger channels and compare each value read with the range of the equipment unit connected to the channel. Halt upon detection of an out of range condition. This provides an independent check on the current stress values being applied to the units under test. - Every five seconds, read current temperatures from temperature chambers and compare each value read with the desired value of the temperature chamber. This assumes that the temperatures have not gone out of control.

L12: Entry 4 of 9 File: TDBD Jul 1, 1992

TDB-ACC-NO: NN9207308

DISCLOSURE TITLE: APL2 Calling C Routines that use Reference Parameters.

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, July 1992, US

VOLUME NUMBER: 35
ISSUE NUMBER: 2
PAGE NUMBER: 308 - 309

PUBLICATION-DATE: July 1, 1992 (19920701)

CROSS REFERENCE: 0018-8689-35-2-308

DISCLOSURE TEXT:

- APL2* and C differ from one another in many respects, one of which is the way their function parameters are used. In APL2, all data is passed by value, whereas it is possible in C to pass data either by value or by reference. APL2 can use C functions by defining them as external functions, but to do so in the general case one must also be able to handle the case where reference parameters are being used, as the C routines called are often pre-defined and cannot easily be changed. This article deals with how this can be done. - Another aspect to take into account is the values that APL2 passes to the C environment. The values are not necessarily copies, but may sometimes be shared between multiple programs. This works fine for output-only parameters, i.e., parameters that will not be updated by the called routine, but does not work at all if an update may occur. - To solve the above two problems, create a temporary buffer large enough to hold the function argument list as well as any data that may be modified. Then store the updatable items in the buffer, starting from the front of the buffer. After this, store the argument list values in the tail end of the buffer. One or more items in the argument list will typically refer back to data stored in the first part of the buffer. These can now be updated at will, as they are copies of the originals. Lastly, call the intended function, passing along the argument list as parameters to the function. After the call has completed and the values passed back to APL2, free the entire buffer in a single step. - An example may be the following. The C function to be called from APL2 has the following function prototype: void fn(int i, char *s, int *io); The function may be called from APL2 as this: fn i s io or using explicit values: fn 1 'first' 2 The buffer built with these values would look like: ***** SEE ORIGINAL DOCUMENT ***** The C function can now update the second and third argument at will, without interfering with the APL2 workspace values themselves. A possible content of the buffer after the function fn has been executed may be: Notice that the content of the character string has been updated, and that the second numeric parameter passed by reference now actually points to a wholly new storage location outside the buffer itself. Depending on what parameters are defined as input/output (or just output, for that matter), the values may now be passed back to APL2. - This solution has several benefits: . It maintains APL2's pass-by-value notions while enabling C routines that use reference parameters to be used. - . It maintains the data integrity of the APL2 workspace data. - . A single C storage allocation and a single free is what is needed to get and release the external

storage. - . Only data marked as updatable by the called routine need be copied. * Trademark of IBM Corp.

Previous Doc Next Doc Go to Doc#

L12: Entry 5 of 9 File: TDBD Apr 1, 1985

TDB-ACC-NO: NN85046584

DISCLOSURE TITLE: Insuring Insert Integrity Over Variable-Length, Spanned Records

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, April 1985, US

VOLUME NUMBER: 27
ISSUE NUMBER: 11
PAGE NUMBER: 6584 - 6587

PUBLICATION-DATE: April 1, 1985 (19850401)

CROSS REFERENCE: 0018-8689-27-11-6584

DISCLOSURE TEXT:

- In certain data base log data sets new data is only added at the end of the data set. Existing data is never updated. This invention describes a method of being able to know if all changes to a data page have been recorded on the non-volatile store without the necessity of synchronously writing the page. The advantage this provides is verification of data integrity without the performance degradation of synchronous page writes as each entry is added to the data base log. In a paging environment data pages are normally on disk storage. When a page must be used or modified, it is brought into the machine's main storage. At some time changed data pages are written back to the disk storage. When a page of data is changed more than once, it is possible that the page of data which exists on the non-volatile disk storage will contain only some of the changes. This happens in the following way: 1. one or more changes are made; 2. another task interrupts and begins execution; 3. the data page used by the first task is removed from main storage and written out to the disk to make room for other data; 4. the first task is later allowed to resume; 5. the first task makes the final changes to the data page; 6. there is a machine failure before the data page with the final changes is written out to the disk. The invention assumes that data pages on the disk storage are initialized to some known value before they are used so that missing data can be detected. The invention also assumes that the machine has some method of copying data in main storage without interruption for a length of N characters. The size of data pages on the disk must be an integral multiple of N. Before each block of data is added to the data base log data set, it is surrounded by control data as follows: ***** SEE ORIGINAL DOCUMENT ***** The first area extends from the next available location to the next integer multiple of N characters (where N is the length of the atomic copy from one main store location to another). The second area is all subsequent complete integer multiples of N. This second area may have a length of zero. The third area is any remaining area less than N characters. The data is copied to the real output destination. The initial area of less than N characters is copied with one copy instruction. Then the groups of N characters on N-character boundaries are copied, N characters at a time. Finally, the last area less than N characters is copied and a "valid entry" flag bit is set in the header of the entry. N-character boundaries are arranged so that an integral number of N-character blocks fit exactly on a data page on the disk. As the data is being copied, it is possible that one or more pages will be written on the non-volatile

store before all the data has been copied to the page. Because of the way the data copy has been done, only three possibilities exist: 1. the initial smaller than N piece of data is missing; 2. an N-character block of data is missing; 3. the final smaller than N piece of data is missing. Because data areas never used are set to the known initial value, any area not written will have that initial value. When the data is read, the first check is for the "valid" flag in the header. If it is not found, then no further checking is required; the entry is not complete. If the "valid" flag is on, then more checking is required. If the first piece of data, up to the first integer multiple of N, is large enough to include the "valid" flag bit, no further checking of this area is required. The valid flag bit was the last change made, and it has successfully been read. However, if the first piece includes only the length fields, these are validated by checking that the forward and backward links do point to other valid entries in the journal and that they in turn point to this entry. Subsequent blocks of N characters on an N-character boundary are the easiest case. Reading N characters of the initial value indicate an error, except in the case of the data being equal to the initial value. This is treated as a special case and rarely occurs. Finally, the validity of the final area less than N characters is checked. Reading the initial value indicates an error since the trailing control information cannot be equal to the initial value. If the final piece is so small that it is only the control field, then it can be checked by verifying that it points back to the start of the current entry. If the data consists of an N-character string of the initial value, the data entry is written to the disk before any valid indicators are set. A special flag is set in the entry indicating N characters of the initial value, the valid flag is set, and the entry is written again.

Previous Doc Next Doc Go to Doc#

First Hit Previous Doc Next Doc Go to Doc#
Generate Collection Print

L12: Entry 6 of 9 File: TDBD Jan 1, 1984

TDB-ACC-NO: NN84014363

DISCLOSURE TITLE: Eavesdrop-Detecting Quantum Communications Channel

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, January 1984, US

VOLUME NUMBER: 26
ISSUE NUMBER: 8
PAGE NUMBER: 4363 - 4366

PUBLICATION-DATE: January 1, 1984 (19840101)

CROSS REFERENCE: 0018-8689-26-8-4363

DISCLOSURE TEXT:

- This communications channel is impossible to eavesdrop on without a high probability of being detected. The security against eavesdropping follows from the uncertainty principle, and would not be compromised by any improvement in technology. The channel consists of a straight black tube about 1 meter in diameter and up to 100 km long through which are sent faint standard pulses of polarized light (0 , 45 , 90 , and 135 polarizations), in a beam with so little divergence (less than 10-5) that transmission losses are only a few per cent. The magnitude of the pulses is such that the receiver, with the best available photomultipliers (about 25 per cent quantum efficiency), detects an expected one photon per pulse. The uncertainty principle prevents an eavesdropper (even one with perfectly efficient photomultipliers) from reliably reading or copying such pulses if he does not know beforehand which pulses are rectilinearly (0 or 90) and which pulses are diagonally (45 or 135) polarized. The sender and receiver do know this information, having agreed on it beforehand (the 'quantum key' information), and can therefore use the pulses to communicate, a 0- or 45-degree pulse standing for a binary zero and a 90- or 135-degree pulse standing for a binary one. Because the receiver's photomultipliers are not perfectly efficient, and because of smaller losses in the quantum efficiency due to beam divergence, losses at the mirrors, etc., about 1/e = 0.368 of the pulses fail to be counted at all. An error-correcting encoding step, described in more detail below, permits the receiver to reconstruct the message despite these lost pulses. On the other hand, if too many pulses are missing, or if the decoder for the error-correcting code reports more than a very few polarization errors among the pulses that were received, the receiver concludes that the message has been subjected to eavesdropping and rejects it. In the drawing, mechanical shutters 10 and movable mirrors 29,31 are used to choose the optical paths in the sending and receiving subsystems, but the actual light pulses should probably be briefer than can be achieved with mechanical means, to reduce the number of spurious counts from the photomultiplier's dark current. This is done by electronically pulsing the laser 14 during the time window when the mechanical shutters 10 are in position (i.e., one open and three closed), and, at the receiving end, discriminating electronically against pulses that arrive outside the briefer electronic time window determined by the pulse time and propagation delay. The neutral density filter 16 in the sending subsystem serves to reduce the intensity of the standard pulses produced by the shutter 10 and polarizer 18 until

they yield an expected one detected photon per pulse for the receiver. The optics of the sending subsystem aims to achieve uniformity of pulse intensity and narrow beam divergence, while that of the receiving subsystem aims to achieve maximum quantum efficiency. As an illustration of how the apparatus would work, assume that the current message bit emanating from the encoder 20 is a 0, and the current quantum key bit is a 1 (signifying diagonal polarization). Then the pulse is sent with the 45-degree shutter open and the other three closed. At the receiving end, the quantum key bit 1 causes the receiver to direct the incoming beam into the 45-135 polarizing beam splitter 22, resulting, typically, in a count in the 45-degree photomultiplier 24, which is then passed to the decoder 26. If a count were instead received in the 135 photomultiplier 28, this would typically be recognized as an error by the decoder 26, and would constitute evidence that the pulse had been subjected to eavesdropping. A beam splitter 23 is provided for use with the 0 photomultiplier 25 and the 90 photomultiplier 27. A transmit/receive movable mirror 29 and a movable mirror 31 for quantum decoding are also provided. With a slightly more complicated apparatus, including extra polarizers and quarter-wave plates, left and right circular polarizations could be used as a third quantum coding mode, on the same footing as rectilinear and diagonal polarizations. ENCODING/DECODING ALGORITHMS A 3-step encoding of the message is used before rendering it into a sequence of polarized light pulses with the help of the quantum key. Given an arbitrary original message M of 1000 bits, let S be the check sum of the addresses of the ones in M. Let M' be the 1020 bit string obtained by concatenating M with S. Let E be an expanded, error-correcting version of M' obtained by applying to M' a nonsystematic random convolutional code (cf. Jelinek, IBMJ. Res. Develop. 13 675-685 (1969)) of rate 1/3 and shift-register length 100. E has length 3360 bits because of the 100 dummy source bits needed to purge the shift registers. A convolutional code with these parameters should be sufficient to reconstruct M' with high probability despite loss of 1/e of the bits of E. In order to render the transmission opaque to the eavesdropper even if he intercepts it, E is now one-time-pad encoded (i.e., bitwise exclusive-ORed) with a random classical key J of length 3360 bits known to the receiver and sender but not to the eavesdropper. The result X=(E xor J) is then encoded into light pulses with the quantum key K, also of length 3360 bits, the polarization of the i'th pulse being 90Xi+45Ki degrees. The receiver uses the quantum key to decide whether to analyze the incoming pulse into rectilinear or diagonal polarizations, exclusive-ORs the result with J to obtain Y, which differs from X chiefly by some pulses being missing. Strictly speaking, Y is a 3360-character sequence over the four-character alphabet: (0,1,n,b), where n = 'neither' signifies a missing pulse and b = 'both' signifies counts in both photomultipliers 24, 28 due to dark current or to a polarization error indicative of eavesdropping (e.g., an erroneously vertically polarized pulse, arriving at a time when the receiver was expecting a 45- or 135-degree pulse, might produce counts in both the 45 and 135 photomultipliers). Decoder 26 attempts to decode Y by sequential decoding of the convolutional code and outputs either the reconstructed message M or else an alarm signifying eavesdropping. This alarm would be triggered by 1) too many missing (n) pulses, suggesting that the eavesdropper diverted some pulses from the beam and did not replace them, or (2) too many (b) pulses or errors to be corrected by the error-correcting code, suggesting that the eavesdropper intercepted some pulses and replaced them with incorrectly forged pulses. The alarm would also be triggered by an inconsistency between the reconstructed M and its check sum field, suggesting a subtler sort of intervention in which the eavesdropper tampered with the message by altering some of the polarization by 90 , e.g., by passing some of the pulses through a sugar solution. Aside from yielding assurance that a message has not been listened to, this scheme allows the key information J and K to be reused for several transmissions, as long as no eavesdropping has been detected, without serious loss of security. This contrasts with the classical situation where a key can be safely used only once (hence the name 'one-time pad'). In particular, a quantum channel can be used in a bootstrapping, or 'self-winding' manner, in which, by the time the original key information has become unsafe, it can be replaced by fresh random key information sent through the channel during the previous key's period of relative safety. KEY

FEATURES 1. The use of narrow beam-divergence free-air (or free-space) optics to send quantum messages over appreciable distances with negligible attenuation. - (If there were more than about an order of magnitude of attenuation, the message could be intercepted, accurately read, and re-injected by an eavesdropper near the high-intensity end of the channel.) 2. The use of error-correcting and classical one-time-pad encoding in combination with quantum coding to make a channel in which eaves dropping and tampering are detectable with high probability, and in which a compromised key can be replaced by a fresh key sent through the channel. 3. The use of polarized photons to encode a message in such a way that it cannot be duplicated by an eavesdropper ignorant of certain key information used in forming the message.

L12: Entry 7 of 9 File: DWPI Apr 6, 2006

DERWENT-ACC-NO: 2006-291044
DERWENT-WEEK: 200630

TITLE: Data formatting and indexing method of database management system, involves converting blocks of bits corresponding to time stamp data, into specific character set, and connecting multiple characters to produce output data string

INVENTOR: GROENVELD, D N

PATENT-ASSIGNEE: GROENVELD D N (GROEI), COMPUTER ASSOC THINK INC (COMPN)

PRIORITY-DATA: 2004US-613952P (September 28, 2004), 2005US-0234568 (September 23, 2005)

Search Selected Search ALL Clear

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
WO 2006036958 A1	April 6, 2006	E	000	G06F017/30
US 20060074971 A1	April 6, 2006		012	G06F017/30

DESIGNATED-STATES: AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KM KP KR KZ LC LK LR LS LT LU LV LY MA MD MG MK MN MW MX MZ NA NG NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SM SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW AT BE BG BW CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IS IT KE LS LT LU LV MC MW MZ NA NL OA PL PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW

APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
WO2006036958A1	September 28, 2005	2005WO-US34572	
US20060074971A1	September 28, 2004	2004US-613952P	Provisional
US20060074971A1	September 23, 2005	2005US-0234568	

INT-CL (IPC): G06F 17/30

ABSTRACTED-PUB-NO: US20060074971A
BASIC-ABSTRACT:

NOVELTY - A time stamp data string of decimal system is selected and converted into binary equivalent comprising multiple bits which are grouped into blocks. Each block is converted into American standard code for information interchange (ASCII) character set, and multiple characters of the character set are connected to produce output data string comprising less characters than time stamp data string,

used for creating an index.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) database management system; and

(2) logic for formatting and indexing data.

USE - For formatting and indexing data for use in database management system (DBMS) (claimed).

ADVANTAGE - Since output data string has less characters than time stamp data string, query response time and database update response time are decreased. Faster query and reliable indexing of database information, are achieved by reducing the number of characters in index.

DESCRIPTION OF DRAWING(S) - The figure shows a block diagram of the data formatting and indexing system.

ABSTRACTED-PUB-NO: US20060074971A
EQUIVALENT-ABSTRACTS:

CHOSEN-DRAWING: Dwg.1/7

DERWENT-CLASS: T01
EPI-CODES: T01-D02; T01-E02C; T01-J05B1; T01-J05B4M;

Previous Doc Next Doc Go to Doc#

L12: Entry 8 of 9 File: DWPI Nov 20, 2002

DERWENT-ACC-NO: 2003-185204
DERWENT-WEEK: 200319

TITLE: Credit card illegal use resisting system

INVENTOR: SUN, H; ZHOU, L

PATENT-ASSIGNEE: ZHOU L (ZHOUI)

PRIORITY-DATA: 2002CN-0104500 (March 25, 2002)

Search Selected Search ALL Clear

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
CN 1380627 A	November 20, 2002		000	G06K005/00

APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
CN 1380627A	March 25, 2002	2002CN-0104500	

INT-CL (IPC): G06F 17/22; G06F 17/30; G06K 5/00

ABSTRACTED-PUB-NO: CN 1380627A
BASIC-ABSTRACT:

NOVELTY - The invention discloses a system for guarding against illegal use of credit cards. The data stored on the credit card includes one or more groups of character strings composed of letters and numbers. Also, a data file stored the characters strings same with the character strings in the credit card are built on the bank that issues the credit card. When the credit card holder is using the card, the host computer in the bank issuing the credit card compares the data character strings stored in the credit card with the data character strings in the database. Whether the comparison is identical ornot makes the decision whether the transaction should be carried out or not. When the transaction is completed, the invented system uses the data cahracter strings stored in the credit card to updatethe host computer in the bank issuing the credit card in random mode automatically.

ABSTRACTED-PUB-NO: CN 1380627A
EQUIVALENT-ABSTRACTS:

CHOSEN-DRAWING: Dwg.0/0

DERWENT-CLASS: T05

L12: Entry 9 of 9 File: DWPI Dec 23, 1987

DERWENT-ACC-NO: 1988-000669
DERWENT-WEEK: 198801

TITLE: Terminal integrated circuit card for bank or store - has CPU and memory which stores data trains from integrated circuit handling system, readable using CPU

INVENTOR: LIJIMA, Y

PATENT-ASSIGNEE: TOSHIBA KK (TOKE)

PRIORITY-DATA: 1986JP-0144114 (June 20, 1986), 1986JP-0144113 (June 20, 1986)

Search Selected Search ALL Clear

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
DE 3720427 A	December 23, 1987		024	
FR 2600444 A	December 24, 1987		000	
US 4839792 A	June 13, 1989		023	

APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
DE 3720427A	June 19, 1987	1987DE-3720427	
US 4839792A	June 18, 1987	1987US-0063653	

INT-CL (IPC): G06F 1/00; G06K 5/00; G06K 19/00; G11C 7/00

ABSTRACTED-PUB-NO: DE 3720427A
BASIC-ABSTRACT:

In the portable integrated circuit a data train is written in a memory when an information signal, indicating the validity of the data train, is added to the data train for storage. When a read-out command from the integrated circuit keyboard system is written into the memory, a CPU coupled to the data memory, adds the validity information signal to the start of the read-out system.

The data sequence is transferred to the keyboard monitor. If such data sequence is to be rewritten into the memory, the CPU adds an information signal indicating a non-validity of the data train. The information signal of the non-validity is converted into a validity indicating information signal after terminating the data sequence writing into the memory.

ADVANTAGE - Easy determination of data train validity at time of first transmission.

ABSTRACTED-PUB-NO: US 4839792A
EQUIVALENT-ABSTRACTS:

When a data string is to be written to the memory, information indicating whether the data string is valid or not is appended to the string, and stored.

When a read instruction is supplied the handling system, a CPU connected to the data memory appends the information, to the start of the readout data string, and supplies the data string to the handling system. When the data string supplied from the handling system is to be written in the data memory, the CPU appends the information indicating invalidity to the data string and writes the data string, thereby updating the information indicating invalidity to information indicating validity after the completion of the data string writing. ADVANTAGE - Data can be reliably invalidated during writing, even when power supply is momentarily cut.

(23pp)

DERWENT-CLASS: T01 T04
EPI-CODES: T01-H01C; T04-K;

Previous Doc Next Doc Go to Doc#

Refine Search

Search Results -

Terms	Documents
L8 and (((fresh$ or updat$) near2 (data or information)) same string$ same (relia$ or trust$ or confid$ or credit$ or integrit$))	4

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: 10/648067

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Monday, March 12, 2007 **Purge Queries** Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; THES=ASSIGNEE; PLUR=YES; OP=OR		
L9	L8 and (((fresh$ or updat$) near2 (data or information)) same string$ same (relia$ or trust$ or confid$ or credit$ or integrit$))	4	L9
L8	l4 or l6 or l7	178	L8
	DB=USPT; THES=ASSIGNEE; PLUR=YES; OP=OR		
L7	(3611316 | 3444523 | 3636519 | 4715399 | 3854905 | 3321159 | 5315510 | 5677955 | 5293895 | 4751640 | 4739478 | 4804825 | 5623547 | 5297032 | 4399831 | 5297031 | 4885685 | 4733531 | 5220500 | 4743278 | 4676463 | 4933842 | 4168718 | 4346442 | 4589526 | 3587062 | 4799156 | 3979005 | 5291398 | 4876648 | 3428946 | 5272623 | 4880185 | 3015461 | 3380029 | 4272257 | 5237500 | 5193056 | 5126936 | 3399383 | 4489745 | 4901762 | 4553565 | 4844278 | 3548381 | 5101353 | 3131789 | 5042751 | 4172680 | 5263329 | 2961643 | 5262942 | 4050608 | 5279323 | 5025372 | 4910676 | 3587057 | 3508578 | 3032746 | 4976398 | 3696343 | 3439340 | 5577121)![PN]	63	L7

L6	("6609114"| "5689650"| "4210961"| "6283412")[PN] *DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; THES=ASSIGNEE; PLUR=YES; OP=OR*	4	L6
L5	l3 *DB=USPT; THES=ASSIGNEE; PLUR=YES; OP=OR*	4	L5
L4	("6609114"| "5689650"| "4210961"| "6283412")[URPN] *DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; THES=ASSIGNEE; PLUR=YES; OP=OR*	111	L4
L3	l1 or L2	4	L3
L2	(location$ or position$ or address$) and (((fresh$ or updat$) near2 (data or information)) same string$ same (relia$ or trust$ or confid$ or credit$ or integrit$)) and (vehic$ or car$ or automobile$) and (map$ or street$ or rout$ or poi$)and @pd<=20020826	2	L2
L1	(location$ or position$ or address$) and (((fresh$ or updat$) near2 (data or information)) same string$ same (relia$ or trust$ or confid$ or credit$ or integrit$)) and (vehic$ or car$ or automobile$) and (map$ or street$ or rout$ or poi$)and @ad<=20020826	4	L1

END OF SEARCH HISTORY

Hit List

First Hit Clear Generate Collection Print Fwd Refs Bkwd Refs
Generate OACS

Search Results - Record(s) 1 through 4 of 4 returned.

☐ 1. Document ID: US 6609114 B1
L9: Entry 1 of 4 File: USPT Aug 19, 2003

US-PAT-NO: 6609114
DOCUMENT-IDENTIFIER: US 6609114 B1

TITLE: System for safe collection of payment including electronic payment receipt generators having electronic purses

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 2. Document ID: US 6283412 B1
L9: Entry 2 of 4 File: USPT Sep 4, 2001

US-PAT-NO: 6283412
DOCUMENT-IDENTIFIER: US 6283412 B1
**** See image for Certificate of Correction ****

TITLE: Anti-slosh liquid propellant tank for launch vehicles

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 3. Document ID: US 5689650 A
L9: Entry 3 of 4 File: USPT Nov 18, 1997

US-PAT-NO: 5689650
DOCUMENT-IDENTIFIER: US 5689650 A

TITLE: Community reinvestment act network

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 4. Document ID: US 4210961 A
L9: Entry 4 of 4 File: USPT Jul 1, 1980

US-PAT-NO: 4210961
DOCUMENT-IDENTIFIER: US 4210961 A

TITLE: Sorting system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L8 and (((fresh$ or updat$) near2 (data or information)) same string$ same (relia$ or trust$ or confid$ or credit$ or integrit$))	4

Display Format: - Change Format

Previous Page Next Page Go to Doc#

Hit List

First Hit | Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 7 of 7 returned.

☐ 1. Document ID: US 20030004963 A1

L4: Entry 1 of 7 File: PGPB Jan 2, 2003

PGPUB-DOCUMENT-NUMBER: 20030004963
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030004963 A1

Y

TITLE: Mapping database users to operating system users in a computer schema

PUBLICATION-DATE: January 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Kagalwala, Raxit A.	Issaquah	WA	US
Wohlferd, David P.	Duvall	WA	US

US-CL-CURRENT: 707/103R; 707/200, 719/310

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☐ 2. Document ID: US 20010025273 A1

L4: Entry 2 of 7 File: PGPB Sep 27, 2001

PGPUB-DOCUMENT-NUMBER: 20010025273
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010025273 A1

TITLE: Parallel data network billing and collection system

PUBLICATION-DATE: September 27, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Walker, Jay	Ridgefield	CT	US
Schneier, Bruce	Oak Park	IL	US

US-CL-CURRENT: 705/77; 705/44, 713/155

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☐ 3. Document ID: US 6856970 B1 0
L4: Entry 3 of 7 File: USPT Feb 15, 2005

US-PAT-NO: 6856970
DOCUMENT-IDENTIFIER: US 6856970 B1

TITLE: Electronic financial transaction system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 4. Document ID: US 6820135 B1
L4: Entry 4 of 7 File: USPT Nov 16, 2004

US-PAT-NO: 6820135
DOCUMENT-IDENTIFIER: US 6820135 B1

TITLE: Modeless event-driven data transformation

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 5. Document ID: US 6816869 B2 0
L4: Entry 5 of 7 File: USPT Nov 9, 2004

US-PAT-NO: 6816869
DOCUMENT-IDENTIFIER: US 6816869 B2

TITLE: Mapping database users to operating system users in a computer schema

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 6. Document ID: US 6684333 B1
L4: Entry 6 of 7 File: USPT Jan 27, 2004

US-PAT-NO: 6684333
DOCUMENT-IDENTIFIER: US 6684333 B1

TITLE: Parallel data network billing and collection system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 7. Document ID: US 5949875 A 0
L4: Entry 7 of 7 File: USPT Sep 7, 1999

US-PAT-NO: 5949875

DOCUMENT-IDENTIFIER: US 5949875 A

TITLE: Parallel data network billing and collection system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L3 and navigat$	7

Display Format: - Change Format

Previous Page   Next Page   Go to Doc#

Refine Search

Search Results -

Terms	Documents
L4 or L5	16

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L6 and (map$ or route or street or intersect$ or poi or navigat$)

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Monday, March 12, 2007 Purge Queries Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; THES=ASSIGNEE; PLUR=YES; OP=OR		
L6	l4 or L5	16	L6
L5	L3 and 707/$.ccls.	12	L5
L4	L3 and navigat$	7	L4
L3	l1 or L2	33	L3
L2	(location$ or position$ or address$) and (((fresh$ or updat$) near2 (data or information)) same string$ same (relia$ or trust$ or confid$ or credit$ or integrit$)) and @pd<=20020826	15	L2
L1	(location$ or position$ or address$) and (((fresh$ or updat$) near2 (data or information)) same string$ same (relia$ or trust$ or confid$ or credit$ or integrit$)) and @ad<=20020826	31	L1

END OF SEARCH HISTORY

Sign in

Google

Web Images Video News Maps **more »**

"freshness information" reliab? vehicle navigat | Search | Advanced Search
Preferences

Web

Did you mean: "freshness information" ***reliable***? vehicle ***navigator***?

No standard web pages containing all your search terms were found.

Your search - **"freshness information" reliab? vehicle navigat?** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.

Google Home - Advertising Programs - Business Solutions - About Google

Hit List

First Hit | Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 10 of 16 returned.

☐ 1. Document ID: US 20030105732 A1

L6: Entry 1 of 16 File: PGPB Jun 5, 2003

PGPUB-DOCUMENT-NUMBER: 20030105732
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030105732 A1

TITLE: Database schema for structure query language (SQL) server

PUBLICATION-DATE: June 5, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Kagalwala, Raxit A.	Issaquah	WA	US
Thompson, John Patrick	Seattle	WA	US

US-CL-CURRENT: 707/1

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 2. Document ID: US 20030018638 A1

L6: Entry 2 of 16 File: PGPB Jan 23, 2003

PGPUB-DOCUMENT-NUMBER: 20030018638
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030018638 A1

TITLE: Full text search system

PUBLICATION-DATE: January 23, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Abe, Fumirou	Kawasaki		JP
Matsuura, Masataka	Kawasaki		JP
Tabata, Yoko	Kawasaki		JP
Nagata, Masahiko	Fukuoka		JP
Hara, Yasuhisa	Fukuoka		JP

US-CL-CURRENT: 707/6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 3. Document ID: US 20030004963 A1

L6: Entry 3 of 16 File: PGPB Jan 2, 2003

PGPUB-DOCUMENT-NUMBER: 20030004963
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030004963 A1

TITLE: Mapping database users to operating system users in a computer schema

PUBLICATION-DATE: January 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Kagalwala, Raxit A.	Issaquah	WA	US
Wohlferd, David P.	Duvall	WA	US

US-CL-CURRENT: 707/103R; 707/200, 719/310

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 4. Document ID: US 20030004935 A1

L6: Entry 4 of 16 File: PGPB Jan 2, 2003

PGPUB-DOCUMENT-NUMBER: 20030004935
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030004935 A1

TITLE: Method and system for long-term update and edit control in a database system

PUBLICATION-DATE: January 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Wilmot, Gerald Johann	Marina	CA	US
Keesey, James Louis	Sunnyvale	CA	US

US-CL-CURRENT: 707/3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 5. Document ID: US 20010025273 A1

L6: Entry 5 of 16 File: PGPB Sep 27, 2001

PGPUB-DOCUMENT-NUMBER: 20010025273
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010025273 A1

TITLE: Parallel data network billing and collection system

PUBLICATION-DATE: September 27, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Walker, Jay	Ridgefield	CT	US
Schneier, Bruce	Oak Park	IL	US

US-CL-CURRENT: 705/77; 705/44, 713/155

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 6. Document ID: US 7080069 B2
L6: Entry 6 of 16 File: USPT Jul 18, 2006

US-PAT-NO: 7080069
DOCUMENT-IDENTIFIER: US 7080069 B2

TITLE: Full text search system

PRIOR-PUBLICATION:

DOC-ID	DATE
US 20030018638 A1	January 23, 2003

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 7. Document ID: US 7054862 B2
L6: Entry 7 of 16 File: USPT May 30, 2006

US-PAT-NO: 7054862
DOCUMENT-IDENTIFIER: US 7054862 B2
**** See image for Certificate of Correction ****

TITLE: Method and system for long-term update and edit control in a database system

PRIOR-PUBLICATION:

DOC-ID	DATE
US 20030004935 A1	January 2, 2003

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 8. Document ID: US 6920455 B1

L6: Entry 8 of 16 File: USPT Jul 19, 2005

US-PAT-NO: 6920455
DOCUMENT-IDENTIFIER: US 6920455 B1
**** See image for Certificate of Correction ****

TITLE: Mechanism and method for managing service-specified data in a profile service

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 9. Document ID: US 6856970 B1

L6: Entry 9 of 16 File: USPT Feb 15, 2005

US-PAT-NO: 6856970
DOCUMENT-IDENTIFIER: US 6856970 B1

TITLE: Electronic financial transaction system

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 10. Document ID: US 6820135 B1

L6: Entry 10 of 16 File: USPT Nov 16, 2004

US-PAT-NO: 6820135
DOCUMENT-IDENTIFIER: US 6820135 B1

TITLE: Modeless event-driven data transformation

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L4 or L5	16

Display Format: - Change Format

Previous Page Next Page Go to Doc#

Hit List

First Hit | Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 11 through 16 of 16 returned.

☐ 11. Document ID: US 6816869 B2

L6: Entry 11 of 16 File: USPT Nov 9, 2004

US-PAT-NO: 6816869
DOCUMENT-IDENTIFIER: US 6816869 B2

TITLE: Mapping database users to operating system users in a computer schema

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 12. Document ID: US 6684333 B1

L6: Entry 12 of 16 File: USPT Jan 27, 2004

US-PAT-NO: 6684333
DOCUMENT-IDENTIFIER: US 6684333 B1

TITLE: Parallel data network billing and collection system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 13. Document ID: US 6615216 B1

L6: Entry 13 of 16 File: USPT Sep 2, 2003

US-PAT-NO: 6615216
DOCUMENT-IDENTIFIER: US 6615216 B1

TITLE: Lock free data structure maintenance

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 14. Document ID: US 6006216 A

L6: Entry 14 of 16 File: USPT Dec 21, 1999

US-PAT-NO: 6006216
DOCUMENT-IDENTIFIER: US 6006216 A

TITLE: Data architecture for fetch-intensive database applications

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. De

☐ 15. Document ID: US 5949875 A

L6: Entry 15 of 16 File: USPT Sep 7, 1999

US-PAT-NO: 5949875
DOCUMENT-IDENTIFIER: US 5949875 A

TITLE: Parallel data network billing and collection system

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. De

☐ 16. Document ID: US 4210961 A

L6: Entry 16 of 16 File: USPT Jul 1, 1980

US-PAT-NO: 4210961
DOCUMENT-IDENTIFIER: US 4210961 A

TITLE: Sorting system

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L4 or L5	16

Display Format: - Change Format

Previous Page Next Page Go to Doc#